








Reto 7: Distribución de vacunas - Corporación Umbrella

 Propósito	Desarrollar una interfaz de consola que permita mostrar los indicadores de desempeño de cada vehículo, para cada cliente
 Dificultad	
 Fecha	@May 23, 2022
 Fuente	MisionTic2022 Ciclo 1
 Numero	26
 Adjuntos	Reto_Semana_6_23may2022.pdf

Autor: **David Santiago Urbano Rivadeneira**

Proceso Ideal

1. **Identificar** el problema

- ¿Cual es el problema? → Desarrollar una interfaz que permita mostrar los indicadores de desempeño de cada vehículo, para cada cliente
- Stakeholders
 - Conductores de Transportes del Norte. (clientes)
 - Departamento de logistica
 - Transportes del Norte
 - Conductores de los camiones
- Objetivos: se detallan en los requerimientos funcionales abajo

Restricciones 

Restricciones

1. Se debe seleccionar la empresa transportadora: Transporte_Norte o Transporte_Monky
2. Seleccionar el numero del camion al que se le calcularan los indicadores de desempeño
3. Desplegar los siguientes indicadores de desempeño para la empresa transportadora

2. Definir el problema

- ¿Que conozco? se ingresaran 2 planillas, Las planillas:
 - *ValoresAsignados.csv* y *ValoresRegistrados.csv* que se ingresara al programa consta de 3 variables (*Punto de distribucion*, *Cajas entregadas* y *Tiempo de despacho(minutos)*).
- Se dividió el problema en los subproblemas siguientes
 - Leer desde el archivo “ValoresAsignados.csv” sus variables correspondientes
 - Leer desde el archivo “ValoresRegistrados.csv” sus variables correspondientes
 - Calcular los 5 indicadores (*Eficiencia en tiempos de despacho (%)*, *Tasa de entrega (cajas/min)*, *Nivel de cumplimiento de los despachos (%)*, *Entregas a tiempo (%)*) para posteriormente imprimirlos en pantalla dependiendo de el punto selecciona de distribucion
 - Imprimir todos los puntos de distribución que registran ambas condiciones negativas: tanto sobre-entregas, como demora en su tiempo de despacho.

3. Estrategia

Ejemplos:

Entrada del programa:

```
Transporte_Monky
3
```

Salida esperada del programa:

```
Eficiencia en tiempo de Despacho = 0.0 %
Tasa de entrega = 7.0 cajas/min
Nivel de cumplimiento = 93.1 %
Entregas a tiempo = 50.0 %
```

4. Algoritmos

Requerimientos funcionales

El aplicativo debera desplegar los siguientes indicadores de desempeño para cualquier punto de distribución:

- Eficiencia en tiempos de despacho (%) = $100 * (\text{Tiempo total de despacho asignado} - \text{Tiempo total de despacho registrado}) / \text{Tiempo de total despacho asignado}$
- Tasa de entrega (cajas/min) = $\text{Cantidad total de cajas despachadas} / \text{Tiempo total de despacho}$
- Nivel de cumplimiento de los despachos (%) = $100 * (\text{Total de cajas despachadas} / \text{Total de cajas asignadas})$
- Entregas a tiempo (%) = $100 * (\text{No. de entregas a tiempo} / \text{No. Total de entregas realizadas})$. Nota: este es un resultado que es constante, es decir, es independiente del punto de distribución porque toma en cuenta todos los puntos de distribución.

Divide y venceras

- Main:
 1. Llamar las funciones
 2. Imprimir los valores solicitados
 3. para las graficas utilizar matplotlib
- Funcion 1 Control:
 1. Almacenar los datos para cada uno de los archivos, recorriendo cada fila y guardandolos en dos nuevas matrices.
 2. Definir las variables necesarias para el calculo de los indicadores, y sus respectivas formulas
 3. Realizar el conteo de cajas entregadas a tiempo

Pseudocodigo para el conteo

Nombre: Cajas entregadas a tiempo

“””Este algoritmo realiza el conteo de cajas entregadas a tiempo“””

INICIO

Variables:

Lista de listas: `tabla_registrados` , `tabla_asignados`

Variable acumuladora: `num_entregas_atiempo`

Recibir: `tabla_registrados` , `tabla_asignados`

```
num_entregas_atiempo = 0
```

```

Para i ← 0 hasta longitud(lista_tuplas_asignado):
    Si tabla_asignados[i][2] < tabla_registrados[i][2] entonces:
        num_entregas_atiempo = num_entregas_atiempo + 1

```

FIN

4. Con los valores anteriores, definimos el iterador en este caso el punto_distribucion ingresado por el usuario. de la siguiente forma:

Empresa_transp	Punto_dist	Id_camion	cajas_entrega	Tiempo_despacho

5. Se realiza la busqueda por los dos parametros y si encuentra el punto la empresa transportadora y el id del camion entonces ahi si realize los calculos de los indicadores
6. el sistema realiza el promedio para variable, puesto que hay varios datos.
7. Se imprime en pantalla los indicadores

Codigo de la solucion



Siguiendo las **convenciones estilísticas** de la PEP8 para escribir codigo en Python mas legible

Codigo Main.py



```
import control as ctr

empresa_transp = input() # con los inputs tuve un problema
id_camion = int(input())

ctr.control(empresa_transp, id_camion)
# ctr.control("Transporte_Norte", 3)
```

Control.py

```

from pathlib import Path

import pandas as pd

def control(empresa_transp: str, id_camion: int) -> str:
    """Lee dos archivos almacenados en data, los convierte a dataframes para posteriormente
    calcular los indicadores y imprimirlos en pantalla"""
    data_path = Path('data')
    # Lecture of csv files
    df_assigned = pd.read_csv(
        filepath_or_buffer=data_path /
        'Valores_Asignados.csv',
        sep=',')
    df_registered = pd.read_csv(
        filepath_or_buffer=data_path /
        'Valores_Registrados.csv',
        sep=',')
    # Search by Empresa_transp and Id_camion
    search = df_assigned.loc[(df_assigned['Empresa_transp'] == empresa_transp) & (
        df_assigned['Id_camion'] == id_camion)]
    search_reg = df_registered.loc[(df_registered['Empresa_transp'] == empresa_transp) & (
        df_registered['Id_camion'] == id_camion)]
    # Check if searched dataframe is empty
    if search.empty or search_reg.empty:
        print("El id del camion o la empresa de transporte no existe")
        return 0
    else:
        filter_assigned = df_assigned.loc[df_assigned['Empresa_transp']
                                           == empresa_transp]
        filter_registered = df_registered.loc[df_registered['Empresa_transp']
                                              == empresa_transp]

    # Variables para calcular los indicadores
    t_total_asignado = search['Tiempo_despacho'].mean()
    t_total_registrado = search_reg['Tiempo_despacho'].mean()
    total_cajas_asignadas = search['cajas_entrega'].mean()
    total_cajas_despachadas = search_reg['cajas_entrega'].mean()
    num_entregas_atiempo = (
        filter_assigned['Tiempo_despacho'] < filter_registered['Tiempo_despacho']).sum()
    # Realizar calculos de los indicadores
    eficiencia = 100 * (t_total_asignado -
                       t_total_registrado) / t_total_asignado
    tasa_entrega = total_cajas_despachadas / t_total_registrado
    nivel_cumplimiento = 100 * \
        (total_cajas_despachadas / total_cajas_asignadas)
    entregas_realizadas = len(filter_assigned)
    entregas_atiempo = 100 * num_entregas_atiempo / entregas_realizadas
    # Output
    print(
        f"Debug info: entregas a tiempo {num_entregas_atiempo} y entregas totales realizadas:
        {entregas_realizadas}")
    print(f"Eficiencia en tiempos de despacho = {eficiencia:.1f} %")
    print(f"Tasa de entrega = {tasa_entrega:.1f} cajas/min")
    print(
        f"Nivel de cumplimiento de los despachos = {nivel_cumplimiento:.1f} %")
    print(f"Entregas a tiempo = {entregas_atiempo:.1f} %")

```

Pruebas


```
*Test_Trans.Monky_3:
exact

Output
1 Eficiencia en tiempo de Despacho = 0.0 %
2 Tasa de entrega = 7.0 cajas/min
3 Nivel de cumplimiento = 93.1 %
4 Entregas a tiempo = 50.0 %

Input
1 Transporte_Monky
2 3
```

```
Eficiencia en tiempos de despacho = 0.0 %
Tasa de entrega = 7.0 cajas/min
Nivel de cumplimiento de los despachos = 93.1 %
Entregas a tiempo = 50.0 %

Process finished with exit code 0
```

 [Link notion](#)